

Method and Apparatus for Efficient and Accountable Distribution of Streaming Media Content to Multiple Destination Servers in a Data Packet Network (DPN)

5

Field of the Invention

The present invention is in the field of video broadcasting and editing, and pertains more particularly to methods and apparatus for
10 distributing media-rich content from a client to multiple destination servers distributed over a Data Packet Network (DPN) for ultimate accessibility by broadband users.

15

Background of the Invention

Media distribution over a data-packet-network (DPN) is accomplished between at least two distributed nodes connected to the network. For example, one node, designated as a sending node typically
20 sends content to a second node designated as a receiving node. Any node may be a sending or receiving node depending on the nature and intent of the transaction. A DPN carries data that is organized into data packets, which are addressed from a sending or *source* node to a receiving or *end* node by data in fields in a header. A DPN comprises all of the lines, connection
25 points and equipment that make up a communications or data transfer network. A good example of a DPN is the well-known Internet network, which will be used as a preferred example throughout this specification.

Data traffic on the Internet may assume the form of a number of varying media types. Some examples are e-mail, IP telephony, electronic
30 information pages, fax messages, voice message, shared files, and so on.

Better techniques for data management and transmission over networks, along with the advent of more powerful processors providing power for connected nodes has recently made transferring video/audio files a practical reality for DPN implementation. The transfer of video/audio content over a DPN is often enhanced with a technique known as multimedia steaming.

Streaming technology involves near real-time data transfer of multimedia files over a data connection or channel set-up between one node and another. Streamed media may be displayed as it downloads and in preferred situations, quality of the media is as good as if downloaded in it's entirety and then played on a display system. Typically, software implemented at both a sending and receiving station in streaming media functions to compress media for sending and to uncompress media for receipt and display of multimedia content. Software media players are provided to enable display of multimedia content wherein the content is audio, video, or a combination thereof.

One problem with current art distribution methods is, that if an error occurs with regard to media receipt at one node, the receiving node must establish a new connection with the source node and begin the download process over again, typically from the beginning. Other problems include managing bandwidth for streaming content. For example, a source server may only transmit at available bandwidth rates supported at the time of transmission. If a dropout occurs there is typically no remedy except to wait for better bandwidth.

One with skill in the art of media transfer, especially over the Internet, will appreciate that there are now many companies and organizations providing multimedia content to end users. End users typically subscribe to offered services and connect to the Internet for the purpose of

receiving streamed content for display on their respective, network-connected nodes.

The technology referred to above ,wherein video/audio content is streamed to end-users, encompasses a wide range of customer equipment, software, and delivery mediums. For example, users may download
5 streaming content to personal computers connected to the network over standard conventional telephone lines. Special digital services such as Integrated Services Digital Network (ISDN) and Digital Subscriber Lines (DSL) make receiving and playing streamed media more practical through
10 faster connections and greater bandwidth. WEB TV, which typically uses a cable delivery method, is another known implementation of the technology.

In a system known to the inventor, media content is collected, pre-processed and distributed to end servers or nodes distributed on a network from which end users may download media-rich content for display and, in
15 some cases (assuming a back channel), interact with while they are connected to the network. These end servers are adapted to serve video/audio presentations on demand to requesting subscriber/users having suitable network connectivity with their equipment.

It has occurred to the inventors that more and more users are taking
20 advantage of new technologies for receiving interactive video/audio stored for access on such as the Internet, for example. As demand increases, it becomes necessary to provide better and more efficient methods for distributing content from a source or sources to an increasing number of participating media-access nodes or servers.

25 What is clearly needed is a method and apparatus that enables fast and accountable distribution of multimedia content over a DPN to multiple user-access nodes distributed on the network. A system such as this would

allow users more access points from which to receive content and provide accountability to media sources as to success or failure of distribution.

5

Summary of the Invention

10 In a preferred embodiment of the present invention a system for efficient streaming of media content from a client content provider to individual Internet destinations is provided, comprising an Internet-connected base server for job initialization and tracking; and a matrix of Internet-connected node servers, at least some of which are to receive the streaming media content. The system is characterized in that the client, the base station and the node servers each execute cooperative software,
15 wherein a client requests a job session of the base server, specifying dimensions of the job, and the base server creates a unique job object defining the job, receives the streaming content from the client, governs distribution of the streaming content to the matrix of node servers according to the job object, and notifies the client content provider of progress and
20 completion.

In a preferred embodiment the base server specifies a server list of all servers to which media content is to be streamed as a part of the job object created. Following creation of a job object at the base server, the base server transmits the job object to at least a first one of the node servers, and
25 then streams the job content to the first node server. The base server then transmits the job object to plural ones of the node servers, and streams the job content to the plural node servers.

In preferred embodiments the first node server, having received the job object, becomes a source server, and transmits the job object and streams the job content to further ones of the node servers. If the first node server having the job object attempts to stream the job content to a second node server already receiving streaming job content from a third node server, and the second node server selects between the first and third node servers for receiving streaming job content according to which of the first and third node servers can transmit at a higher rate.

The client content provider, having established a first job object at the base server and having begun streaming job content to the base server, may establish a second job object at the base server for distributing second streamed job content to the node servers. There may also be plural client content providers, wherein the base server establishes job objects for individual ones of the plural client content providers and streams job content for plural job objects to plural downstream nodes simultaneously.

In preferred embodiments of the invention, as each node server completes a job, notification of completion is passed to the base server, which updates the client content provider of progress made. Also in some embodiments, after all node servers have completed a job and have reported to the base server, the base server closes the associated job object and updates the client content provider associated with the job object.

In some embodiments the base server tracks completion of the job by the matrix of node servers, and if the job is not completed by all node servers within a set time, the base server queries the node server or servers failing to complete. Receiving no response from a node server, the base server may attempt to repair the node server not responding.

In another aspect of the invention a method for efficient streaming of media content from a client content provider to individual Internet

destinations is provided, comprising steps of (a) requesting, by the client content provider of a base server, creation of a job object defining the job by dimensions supplied by the client content provider; (b) creation, by the base server, the job object requested; (c) receiving, by the base server, streaming
5 job content from the client content provider according to the job object; (d) distributing, by the base server, the job object and streaming job content to individual ones of plural node servers at the Internet destinations; and (e) notifying the client content provider, by the base server, of progress and completion of the job according to the job object.

10 In some embodiments, in step (b), the base server specifies a server list of all servers to which media content is to be streamed as a part of the job object created. Also, in (d) following creation of a job object at the base server, the base server may transmit the job object to at least a first one of the node servers, and then stream the job content to the first node server
15 The base server may also transmit the job object to plural ones of the node servers, and stream the job content to the plural node servers.

In a preferred embodiment a first node server, having received the job object and streaming job content, becomes a source server, and transmits the job object and streams the job content to further ones of the node
20 servers.

In some cases a first node server having the job object attempts to stream the job content to a second node server already receiving streaming job content from a third node server. In this case the second node server may select between the first and third node servers for receiving streaming
25 job content according to which of the first and third node servers can transmit at a higher rate.

In alternative embodiments the client content provider, having established a first job object at the base server and having begun streaming

job content to the base server, may establish a second job object at the base server for distributing second streamed job content to the node servers. There may also be plural client content providers, and steps for the base server establishing job objects for individual ones of the plural client content providers and streaming job content for plural job objects to plural downstream nodes simultaneously.

In some embodiments of the method there steps for notification of completion by each node server associated with a job as each node server completes the job being passed to the base server, and the base server updating the client content provider of progress made. After all node servers have completed a job and have reported to the base server, the base server closes the associated job object and updates the client content provider associated with the job object.

In some embodiments, if the base server tracking completion of the job by the matrix of node servers notes the job is not completed by all node servers within a set time, the base server queries the node server or servers failing to complete. The base server may also attempt to repair failing node servers.

In embodiments of the present invention, taught in enabling detail below, for the first time an efficient streaming distribution system is provided for streaming media from content providers to multiple Internet-connected servers.

Brief Description of the Drawing Figures

Fig. 1 is an overview of a media-distribution system practiced on a DPN according to an embodiment of the present invention.

Fig. 2 is a block diagram illustrating components of a job object according to an embodiment of the present invention.

Figs 3-8 are block diagrams illustrating job/command distribution and notification levels according to an embodiment of the present invention.

5 Fig. 9 is a block diagram illustrating a job end notification according to an embodiment of the present invention.

Fig. 10 is a block diagram illustrating an incomplete job completion notification according to an embodiment of the present invention.

10 Fig. 11 is a block diagram illustrating a server recovery and reactivation protocol according to an embodiment of the present invention.

Description of the Preferred Embodiments

15 According to a preferred embodiment of the present invention a streaming media-distribution system is provided for distributing streaming multimedia content and server commands to a plurality of distributed media servers.

20 Fig. 1 is an overview of a streaming media-distribution system 9 practiced on DPN 11 according to an embodiment of the present invention. Media-distribution system 9 is practiced, in this example, on the well-known Internet network, which is a DPN and will hereinafter be referred to as Internet 11. Any other DPN may be used in place of or in combination with
25 Internet 11 to practice the present invention. The inventor chooses the Internet network because of its high public-access characteristic.

An Internet backbone 19 is illustrated within Internet 11 and represents all of the lines and connection points making up the Internet

network in a global sense. Therefore, there are no geographic limits for practicing the present invention. Furthermore, sub-nets connected to Internet 11 may be included in the scope of media-distribution system 9.

A client station 13 is provided within Internet 11 and illustrated as
5 connected to backbone 19. Client station 13 represents any computer-enabled entity connected to Internet 11 that functions as a source for distributing streaming media content over Internet 11. Client 13 may be a server capable of issuing pre-programmed commands and initiating timed distribution of streaming media content.

10 In another embodiment, client 13 may be a manned computer node or system of nodes adapted for streaming media and server-command distribution. In this case, client 13 is within Internet 11 and is presumed to be an ever-connected part of Internet 11. However, client 13 may be a remote station that connects to Internet 11 through dial-up means or other
15 means known in the art for accessing the Internet.

A second client station 15 is provided within Internet 11 and is illustrated as connected to backbone 19. Client 15 may be assumed to possess all of the attributes which were described above concerning client 13. It is noted here that clients 13 and 15 may differ from each other within the
20 range of the description of client 13, which also applies equally to client 15.

A base station (BS) 14 is provided within Internet 11 and illustrated as connected to backbone 19. BS 14 is enabled as a main or controlling central server that functions as an interface between clients 13 and 15 and a plurality of end servers (S1-Sn) illustrated as distributed within Internet 11 and connected to backbone 19. Servers S1-Sn are enabled as media servers
25 that are capable of delivering streaming media to end users as well as being adapted for receiving and distributing streaming media from and to other like media servers provided within Internet 11 (individual ones of S1-Sn).

Servers S1-Sn are slaved to BS 14 in a preferred embodiment such that they may be controlled in some instances by BS 14. However, servers S1-Sn may also function independently from BS 14 according to a media-distribution protocol practiced on network 9.

5 It will be apparent to one with skill in the art that there may be any number of servers S1-Sn distributed within network 9 than are physically illustrated in this example without departing from the spirit and scope of the present invention.

10 A user premise 21, also labeled as user premise 1, is illustrated as connected to network 9 within Internet 11 by virtue of an Internet access line 27. Premise 21 is adapted, as illustrated by a monitor-display icon included therein, for connecting to Internet backbone 19 and communicating with any one of servers S1-Sn for the purpose of downloading streaming media content. Internet access line 27 represents any known means for
15 Internet access. Examples include but are not limited to dial-up services as offered through an Internet Service Provider (ISP), cable/modem service, wireless satellite connection, and so on. In the case of a land-line connection, an ISP and telephone network are not illustrated for the purpose of simplicity, but they may be assumed to be present. In the case of a
20 wireless access means, a wireless network is not illustrated for the same reasons described above, but may be assumed to be present.

 A user premise 23, also labeled as user premise n, is illustrated as connected to network 9 within Internet 11 by virtue of an Internet access line 25. User premise 23 is adapted for communicating with and receiving
25 media content from any one of servers S1-Sn as was described with user premise 21. Internet access line 25 represents any known means of Internet access as described with line 27. It will be appreciated by one with skill in the art that there may be many more user premises connected to backbone

19 and communicating with servers S1-Sn than are physically illustrated in this example without departing from the spirit and scope of the present invention.

An object of the present invention is to provide a distribution
5 network (9) wherein streaming media may be distributed by a client (13, 15) to end nodes (S1-Sn) in a more efficient and accountable fashion than is available in prior art systems. To accomplish this object the inventor provides a software component termed a Job object, two of which, J1 and Jn, are illustrated as executing on BS 14 in Fig. 1, collectively labeled with
10 the element number 16. A job object 16 defines a job for distributing media to one or more of servers S1-Sn. One job object 16 is created for every media-distribution task defined and initiated by a client (13, 15). In this example, J1 may represent a job initiated by client 13 and Jn may represent a job initiated by client 15. Job objects 16 are created on BS 14 in response to
15 client initiation of a job for media distribution to one or more of servers S1-Sn.

Job objects 16 are, in a preferred embodiment, Java-based, executable program-lists that provide instruction to servers S1-Sn for handling and redistributing streaming media. It is noted herein that any
20 participating node must support the protocols of the system and must understand and be able to work with job objects, which function is provided by software executing on each such node. Each job object is unique to a client and a job the client wishes to accomplish. More detail about job objects is provided further below.

25 In practice of the present invention, a client, say client 13 for example, contacts BS 14 to establish a "job-distribution session". BS 14 creates a job object, object 1 for example, on behalf of client 13. Job object 1 is a software entity and contains all of the data required to facilitate a job

along with a list of servers among servers S1-Sn for data and command distribution during a job. Client 13 remains connected to BS 14 during job and command distribution and processing and may add commands to the job as distribution proceeds through network 9.

5 The method of distribution is such that the job object carries command instruction along with streaming media content, which is distributed to at least a first server, say server S1 for exemplary purposes. Server S1 then attempts to distribute the job object and the media content to other servers listed in the job object. Secondary servers then distribute to
10 third-layer servers and so on until all of the servers listed in job object 1 receive their jobs instructions and media content.

 As each server S1-Sn receives instruction and content, they are transformed into source servers that continue distribution. As each server completes a job, which includes distribution to still more servers (if listed), it
15 sends notification thereof back to BS 14. When a client is finished with a job session, the session may be terminated with BS 14. However, BS 14 will continue to monitor job progress until all of the servers have responded with job-completion command receipt notification and have actually completed all of their job processes. At close, each server will respond back to BS 14 that
20 a particular job is completely executed. Any server that does not report back to BS 14 at the close of a job may be queried by BS 14 as to current status. If a server has malfunctioned, then repair may be effected to the malfunctioning server by BS 14.

 By practicing the present invention, which includes observance of
25 various protocols, which are described below, a client may accomplish streaming of media content to a plurality of end servers in an optimized fashion where transfer rates are concerned and may also recover services of end servers that have malfunctioned under certain circumstances.

Fig. 2 is a block diagram illustrating components of job object (16) of Fig. 1 according to an embodiment of the present invention. Job object 16, further defined by plural objects J1-Jn of Fig. 1, has a server address list 29 incorporated therein and adapted to include server addresses of all of the target end-servers that media will be streamed to. A server address defines a permanent network location for a server.

A job command queue 31 is provided within object 16 and is adapted to contain commands issued by a client for furthering the distribution process. Queue 31 may be added to in the field while a distribution process is active by virtue of an open and active session established between a client and a BS. Commands are distributed to end servers in the same way that the job objects are distributed.

Object 16 also has a job identification field 33 provided therein and adapted to contain a job identification number or name, which specifically identifies the job and the client, and is unique to a client. A command identification field is provided within object 35 and adapted to contain command ID numbers specific to commands contained in queue 31. Object 16 also contains a base server address field 37 adapted to contain the address of the originating BS for notification purposes.

As described with reference to Fig. 1, job objects 16 are created at BS 14 on behalf of clients 13 and 15, also of Fig. 1. Referring back to Fig. 1, for exemplary purpose, client 15 sends a request to BS 14 to initiate a media-distribution job. BS 14 may either accept or deny the request. If the request is accepted, BS 14 creates job object 16 and assigns a unique job identification number represented in Fig. 2 by element 33. BS 14 also provides a list of all of the participating servers present on network 9 (Fig. 1). The aforementioned information, field value 33 and list 29, is then sent back to client 15. BS 14 then idles in session waiting to receive commands

from client 15. Such commands may originate at client 15 if it is a manned station, or may come from a remote user connected to client 15 if it is an unmanned server.

Commands and associated data are sent to queue 31 from client 15 using a Command Transfer Protocol (CTP), which is similar to File Transfer Protocol (FTP), a well-known and commonly used data transfer protocol. It is noted herein that a command may not necessarily accompany data. This protocol allows, among other things, controlled file uploads and downloads between servers. The protocol also insures that file attributes are copied.

Other attributes of CTP include the ability to browse a remote directory and enumerate sub-directories and existing files located in them. It is also possible with CTP to obtain file and directory properties from specific files and directories found on remote servers, and to change such properties. New files and directories may also be created using CTP.

It is noted herein that each distributed command coming from a client has three operative states. These are *new* (pending execution), *in progress* (currently being executed), and *completed* (finished executing). More detail about distributing commands is provided in the Figs. below.

Figs. 3-8 are block diagrams illustrating job/command distribution and notification levels according to an embodiment of the present invention. Referring now to Fig. 3, client 13 has requested and been granted a session with BS 14, and job object 16 has been created. This is now Job 1 (J1). The session request is illustrated herein by a directional arrow leading from client 13 and progressing to BS 14. A job ID and server list is sent back to client 13 after job object 16 is created as illustrated by a dotted directional arrow leading from BS 14 and progressing toward client 13. In this example, S1 is the first server to which information and media will be distributed as illustrated by a directional arrow leading from BS14 to S1.

Referring now to Fig. 4, it is noted that S1 now has a copy of J1 (16) that it has received from BS14. This includes any commands sent to BS14 in the interim. Such commands are held in queue 31 of Fig. 2 as previously described. S1, upon receiving the job information, commands and media
5 from BS14, sends notification (illustrated by a dotted directional arrow) back to BS14. At this point, notification information is passed back to client 13 (dotted arrow between BS14 and client 13). If server S1 is the only server in list 29 (Fig. 2), then it may complete its function without further distribution. In actual practice, however, there will likely be many more
10 down-line servers to which data and media will be distributed.

To illustrate a command notification, assume that client 13 has added a new command to command queue 31 in J1, and that command is distributed to S1 and is logged into command queue 31 at S1. The command state is set to *new* and does not change until S1 begins execution
15 of the command. Therefore, there are three notifications that may be sent back to BS14 and ultimately to client 13. The first notification is that S1 received the new command. The second notification is that S1 is executing the command. The third notification is that S1 has completed execution of the command. All three notification states may be represented by the dotted
20 directional arrows leading from S1 back to client 13. However, the first two notifications are solicited, in preferred embodiments, by query if the information is desired. The final notification (completion) is automatic every time a command is completely executed. S1 is in this example distributing job information (J1). Subsequent commands concerning J1, which are sent
25 by client 13, are distributed in the same fashion as job information.

In this example, S1 attempts to distribute job information to S2, which is included in server address list 29 (Fig. 2). It is noted herein that S1 may attempt to distribute to more than one listed server simultaneously. For

exemplary purposes however, S1 is illustrated as attempting to distribute only to S2.

Referring now to Fig. 5, S2 now has J1 from S1 and breaks connection with S1. In this case, both S1 and S2 are now source servers, who attempt to further distribute media and command information to subsequent servers. S2 sends job receipt notification back to BS 14 as illustrated by the dotted directional arrow leading from S2 back to BS14. BS14 passes the information back to client 13 as illustrated by the dotted directional arrow leading from BS14 back to client 13. S1 is now attempting to distribute job information to S5, which is included in field 29 (Fig. 2). S2 is attempting to distribute to S4 representing distribution and notification at a 2nd layer. It is important to note herein that automatic notifications may only comprise job receipt notifications and command completion notifications. However, other types of status notifications may be made automatic by programming if so desired.

Referring now to Fig. 6, S4 and S5 have received job information from S2 and S1 respectively. S4 and S5 send notification directly to BS14 as illustrated by the dotted directional arrows leading from both S4 and S5 and progressing toward BS14. Notification is then passed to client 13 as previously described. S1, S2, S4, and S5 are now all source servers attempting to distribute information to other servers on list 29 (Fig. 2), if any.

It is illustrated in this example that S2 is attempting to distribute job information to S3 and S5 is attempting to distribute job information to S_n as illustrated by a directional arrows, one leading from S2 and progressing toward S3 and one leading from S5 and progressing toward S_n. This represents third layer distribution and notification. Each time a participating

server receives job information and subsequent commands, it sends notification thereof back to BS14.

5 It will be apparent to one with skill in the art that the distribution protocol of attempting to distribute to all down-line servers, and the function of breaking connection with BS14 to act as a new source server, causes distribution frequency to increase exponentially with greater numbers of participating servers. As previously described, commands are distributed in the same manner as job information and are processed by each receiving server with that server attempting to distribute the same command to all of
10 the other servers on list 29 (Fig. 2).

Referring now to Fig. 7, S1 attempts to distribute information to S3 as illustrated by a directional arrow leading from S1 and progressing toward S3. However, S3 is currently receiving the same information from S2 as illustrated by a dotted directional arrow leading from S2 and progressing
15 toward S3. In this case, a dynamic switch may occur if it is determined by S3 that S1 has a faster data transfer rate than S2. Dynamic Switching Optimization (DSO) allows a receiving server to switch to another sending server attempting to send data if the alternative sending server transfers at a higher data rate. Assuming this is the case, S1 will take over the data
20 transfer at the point where S2 left off and S2 will move on to attempt distribution to a next server. More about this and other optimization techniques is detailed later in this specification.

S4 is idle in this example meaning that it has not attempted to distribute its job information or media to any other servers. This may be due
25 to an error or malfunction in server hardware or software. A recovery technique for querying status of S4 and recovering its services before or after termination of a media-distribution job is described further below.

Referring now to Fig. 8, S3, and Sn have received job information from S1 and S5 respectively, and send notifications thereof to BS14 as illustrated by dotted directional arrows, one leading from S3 and progressing toward BS14, and one leading from Sn and progressing toward BS14. This represents a state wherein all servers on list 29 (Fig. 2) have received job information and final notifications are arriving to BS14. Again, commands are propagated in the same fashion, and notification of completed commands are sent back to BS14 directly from executing servers. After all of the target servers receive and execute all commands and distribute their media, final completion notifications are sent to BS14 from participating servers. This occurs after client 13 has sent end-job commands as further described below.

Fig.9 is a block diagram illustrating a job end notification according to an embodiment of the present invention. In this example, it is assumed that all job information and commands have been distributed to all of the participating servers S1-Sn. At this point, a client sends notification of ending a job as illustrated by a directional arrow leading from client 13 and progressing toward BS14. The end-job command is distributed to the first server (S1) and subsequently distributed to all of the other servers as logically illustrated by dotted directional arrows connecting the servers. In actual practice, servers will break off and become source servers and attempt to distribute notification to all of the other servers in the same manners as described above, including dynamic switching where appropriate.

At this point, client 13 will end session with BS14. However BS14 must wait for all notification to come in from all of the affected servers regarding completion of their functions defined by successful execution of all of their remaining commands. Optimization protocols are provided by the inventor to optimize command and data distribution. These protocols are listed as follows:

1) **Command Propagation Optimization:** Whenever a server propagates a command, a list of indexes is also sent to the destination server for that command. Each index value in the list points to a server in the server list, that was passed to a target server during Job creation. These are servers that
5 were not approached by a source server for propagation of a command untill that time. The target server will try to distribute the command only to these servers in list.

2) **Multiple Node Handling:** A Server may open connections with multiple down-line servers to distribute the streaming media content and
10 commands. It may process these connections simultaneously. The number of down-line servers it negotiates with may be governed by a number of factors, like optimum use of available bandwidth, processing power and system wide resources.

3) **Data Rate Sort Order:** Each server may sort its list of servers in
15 descending order of the data-upload rate they support. Thus, the servers will now be addressed in the descending order of their upload data rates. It is noted herein that this may not always be the fastest way of carrying out the media distribution process from the point of view the overall system. However, sorting by data rate will, in most cases, achieve a better efficiency
20 for the system overall, when compared to a random selection of servers from the list.

Dynamic Switching: Whenever a server with a Job ID and a command ID requests services with a subsequent server for the purpose of distributing streaming media content and commands, it will also inform the receiving
25 server of the data rate that it can support for the current transfer. This mechanism comes into play when that command ID is already being served at the destination server by another server in the system. The destination server can compare the data rate supported by the current source server with

the data rate of the requesting server, which proposes to serve the same data. In a case wherein better efficiency may be achieved with the requesting server, the destination server can dynamically switch to the new server.

Fig. 10 is a block diagram illustrating an incomplete job completion notification according to an embodiment of the present invention. In this example, client 13 has sent an end-job command as illustrated in Fig. 9 and has now quit its session with BS14 and is no longer connected. However, BS14 is still receiving final notifications from servers S1, S2, S3, S5, and Sn. This process occurs during a wait period illustrated herein as a time period from T0 to Tn illustrated immediately above BS14. It is noted herein that during time period T0 to Tn, that server S4 has not reported back that it has completed its stated function. It may be that S4 is still processing final commands. It may be that server S4 is malfunctioning as illustrated in this example. Even though client 13 is no longer in session with B.S 14, BS14 will call client 13 to give final notification status after all servers have reported back.

Before final notification is sent to client 13, BS14 will query the status of S4 in an attempt to determine its operating status. If a status report indicates that final commands are still in progress or new and not yet in progress, then wait period T0 to Tn will be extended for another unit of time. If S4 completes execution and notification within the extended period, then BS14 will declare the job completed and will notify client 13. It is noted herein that in some cases this process may take more than one or two wait periods. If there is no response or there is an error response sent by S4 as a result of a status query sent by BS14, then it may be inferred that there is an application error that has occurred or is occurring either before or during execution or, that S4 is unavailable or off-line due to hardware malfunction.

Fig. 11 is a block diagram illustrating a server recovery and reactivation protocol according to an embodiment of the present invention. In this example it is assumed that S4 of Fig. 10 was malfunctioning and did not respond with completion notification during T0 to T1 (1st wait mode). A status query illustrated by a directional arrow labeled I (query) is sent to S4 by BS14 during a second wait mode. S4 responds with an error notification illustrated as a directional arrow labeled II (status response). Assuming that response II indicates an on-line status but failing in final execution of a command or commands, an attempt to repair S4 is initiated by BS14 as illustrated by a directional arrow labeled III (Repair Server). A subsequent initiated communication from BS14 attempts to recover services as illustrated by a directional arrow labeled IV (Activate Server Status).

If S4 is successfully reactivated, then it may proceed to complete its function and send final notification thereof back to BS14 during an extended wait time. Once finished, BS14 will officially close the job and send final notification back to client 13 as illustrated by the dotted directional arrow labeled notify client. In some cases, such as hardware failure or the like, BS14 may notify client 13 of the situation and an expected wait period for bringing server S4 back on-line.

It will be apparent to one with skill in the art that the various configurations illustrated above with respect to Figs. 3-11 represent very simple example-states only. In actual practice of the present invention, base to server, server to server, and server to base transactions occur at a rapid rate according to all of the optimization rules described above. Further, although various states for propagation of a single job object have been described in detail, it is not necessary that a first job be completed and verified before a second job object is created and begins propagation. Such

as second, and subsequent jobs may initiate with client 13, client 15, or any other client in the system.

5 It will also be apparent to one with skill in the art that the method and apparatus of the present invention may be practiced on the Internet network, an Intranet network, or any other DPN adapted to support the required protocols without departing from the spirit and scope of the present invention. Therefore, the methods and apparatus of the present invention should be afforded the broadest possible scope. The spirit and scope of the present invention is limited only by the claims that follow.

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2